Listing of Claims

1. (Currently Amended) A method for executing an operation upon a linked data structure having at least one element, the method comprising the steps of:

(a) performing a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions;

(b) developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions and being associated with a set of pointers to the linked data structure, the set of pointers being stored external to the linked data structure, the second set of element state transitions being distinct from the first set of element state transitions; and

(c) performing the second set of operation tasks in a second phase using the set of pointers.

2. (Original) The method of claim 1 wherein the first set of operation tasks includes navigating existing data structure links.

3. (Currently Amended) The method of claim 1 wherein the step of developing a second set of operation tasks further comprises developing the set of pointers to the data structure~~, the pointers being used in the step of performing the second set of operation tasks in a second phase~~.

4. (Currently Amended) The method of claim 1 wherein the first phase comprises performing parallel operations on the linked data structure and the second phase comprises performing serial operations on the linked data structure, each of the serial operations being developed during one of the parallel operations~~operation tasks of the second set of operation tasks are performed atomically~~.

5. (Currently Amended) The method of claim 1 wherein the ~~step of developing a second set of operation tasks further comprises developing the second set of operation tasks as~~ set of pointers is stored in a list.

6. (Currently Amended) The method of claim 5 wherein the ~~list further comprises~~set of pointers is stored in a first in last out list.

7. (Original) The method of claim 1 wherein the step of developing a second set of operation tasks further comprises a step of performing a conflicts check for the operation.

8. (Original) The method of claim 1 wherein the first set of element state transitions further comprises:

(a) a valid state to a pending delete state transition;

(b) a pre-associated state to a pending insert state transition; and

(c) a pending insert state to a hidden state transition.

9. (Original) The method of claim 1 wherein the second set of element state transitions further comprises:

(a) a pending insert state to a valid state transition;

(b) a pending delete state to an invalid state transition;

(c) a hidden state to an invalid state transition;

(d) a pending delete state to a valid state transition;

(e) a hidden state to a pending insert state transition; and

(f) a pending insert state to an invalid state transition.

10. (Original) A method for performing insertion and deletion operations on elements in a linked data structure, the method comprising the steps of:

(a) performing a first set of operation tasks in a first phase for each insertion and deletion operation, the first set of operation tasks operable to effect a first set of element state transitions;

(b) developing a second set of operation tasks for each insertion and deletion operation, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and

(c) performing the second set of operation tasks in a second phase.

11. (Currently Amended) A method for executing operations upon a linked data structure having at least one element, the method comprising the steps of:

(a) queuing operation tasks in a task queue;

(b) receiving the queued operation tasks;

(c) performing a first set of the operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions;

(d) developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and

(e) performing the second set of operation tasks in a second phase.

12. (Currently Amended) The method of claim ~~11~~47 ~~further comprising repeating steps (a) through (e)~~wherein the first part of each of the operations are performed in parallel and the second part of each of the operations is performed in series.

13. (Currently Amended) A method of inserting a plurality of~~n~~ elements into a linked data structure comprising the steps of:

(a) performing a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions including a pre-associated state to a pending insert state transition for each of the plurality of elements;

(b) developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions including a pending insert state to a valid state transition for each of the plurality of elements; and

(c) performing the second set of operation tasks in a second phase.

14. (Currently Amended) The method of claim 13 wherein the pre-associated state to a pending insert state transition is accomplished by:

(a) marking ~~the~~ an element to be inserted as being pre-associated to the data structure;

(b) navigating the data structure to an insertion point;

(c) creating links between the element to be inserted and the data structure at the insertion point, the links created being visible only to the insertion operation; and

(d) marking the element as being pending insert.

15. (Original) The method of claim 14 wherein the pending insert state to a valid state transition is accomplished by:

(a) creating instructions for making the created links visible to all operations; and

(b) creating instructions for making existing links at the insertion point invisible to all operations.

16. (Original) The method of claim 15 wherein the step of performing the second set of operation tasks further comprises executing the created instructions including marking the element as valid.

17. (Currently Amended) A method of deleting ~~an~~ a plurality of elements ~~into~~ from a linked data structure comprising the steps of:

(a) performing a first set of operation tasks in a first phase, the first ~~set of operation tasks operable to effect~~phase including ~~a first set of element state transitions including~~ a valid state to a pending delete state transition for each of the plurality of elements;

(b) developing a second set of operation tasks, ~~the~~ for execution in a second ~~set of operation tasks operable to effect a second set of element state transitions~~ phase including a pending delete state to an invalid state transition for each of the plurality of elements; and

(c) performing the second set of operation tasks in ~~a~~ the second phase.

18. (Original) The method of claim 17 wherein the valid state to a pending delete state transition is accomplished by:

(a) navigating the data structure to a deletion point;

(b) creating links at the deletion point visible only to the deletion operation; and

(c) marking the element to be deleted as pending delete.

19. (Original) The method of claim 18 wherein the pending delete state to an invalid state transition is accomplished by:

(a) creating instructions for making the created links visible to all operations; and

(b) making existing links at the deletion point invisible to all operations.

20. (Currently Amended) The method of claim 19 wherein the step of performing the second set of operation tasks further comprises executing the instructions including marking ~~the~~ members of the plurality of elements to be deleted as invalid.

21. (Original) A method for executing an operation upon a linked data structure having at least one element, the method comprising the steps of:

(a) grouping a first plurality of operation tasks of the operation in a first set of operation tasks, the first set of operation tasks operable to effect a first set of element state transitions;

(b) performing the first set of operation tasks in a first phase;

(c) grouping a second plurality of operation tasks of the operation in a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and

(d) performing the second set of operation tasks in a second phase.

22. (Currently Amended) A method for executing ~~an~~ operations upon a linked data structure having at least one element, the method comprising the steps of:

(a) creating first and second sets of operation tasks, the first set of operation tasks being characterized by navigation of the linked data structure using at least an existing link, and the second set of operation tasks being distinct from the first set of operation tasks and each being characterized by at least a different pointer from outside the linked data structure to the linked data structure; and

(b) performing the first set of operation tasks in a first phase and the second set of operation tasks in a second phase.

23. (Currently Amended) A method for executing ~~an~~ a plurality of operations upon a linked data structure having at least one element, the method comprising the steps of:

(a) dividing each of the plurality operations into first and second distinct sets of operation tasks;

(b) performing the first set of operation tasks of the plurality of operations together in a first phase; and

(c) performing the second set of operation tasks of the plurality of operations together in a second phase.

24. (Currently Amended) The method of claim 23 wherein the first set of operation tasks is operable in parallel to maintain the linked data structure in an existing linked state.

25. (Currently Amended) The method of claim 24 wherein the second set of operation tasks are operable in series to modify the existing linked state.

26. (Currently Amended) The method of claim 23 wherein each of the first set of operation tasks is visible only to one of the plurality of ~~the~~ operations.

27. (Currently Amended) The method of claim 26 wherein the second set of operation tasks is visible to each of ~~a~~ the plurality of operations.

28. (Currently Amended) A system for executing an operation upon a linked data structure having at least one element, the system comprising:

(a) a memory for storing the linked data structure;

(b) a processor coupled to the memory, the processor operable to perform a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions, to develop a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions, and to perform the second set of operation tasks in a second phase by using a plurality of pointers stored external to the linked data structure to navigate to a plurality of nodes within the linked data structure.

29. (Currently Amended) A system for executing an operation upon a linked data structure having at least one element, the system comprising:

(a) a first memory space for storing the linked data structure;

(b) a second memory space for storing a plurality of pointers separate from the linked data structure; and

(c) a processor coupled to the first and second memory and operable to

divide the operation into first and second distinct sets of operation tasks,

perform the first set of operation tasks in a first phase,

develop a second set of operation tasks in the first phase, development of the second set of operation tasks including storing at least one pointer to the linked data structure in the second memory space, and

perform the second set of operation tasks in a second phase using the at least one pointer.

30. (Currently Amended) A computer readable medium for executing ~~an~~ a plurality of operations upon a linked data structure having at least one element, the computer readable medium comprising:

(a) a code segment for performing a plurality of first phase operation tasks in a first phase, each of the plurality of operation tasks being associated with a different member of the first phase operation tasks;~~a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions~~;

(b) a code segment for developing a plurality of second phase operation tasks during the first phase, each of the plurality of operation tasks being associated with a different member of the plurality of second phase operations~~a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions~~;

(c) a code segment for performing the plurality of second phase operations tasks in a second phase after completion of the first phase~~the second set of operation tasks in a second phase~~.

31. (Cancelled)

32. (Original) A system for executing an operation upon a linked data structure having at least one element, the system comprising:

(a) a means for performing a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions;

(b) a means for developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, and being associated with a set of pointers to the linked data structure, the set of pointers being stored external to the linked data structure, the second set of element state transitions being distinct from the first set of element state transitions; and

(c) a means for performing the second set of operation tasks in a second phase using the set of pointers~~performing the second set of operation tasks in a second phase~~.

33. (Currently Amended) A system for executing ~~an~~ operations upon a linked data structure having at least one element, the system comprising:

(a) a means for performing a first part of each of the operations in a first phase by navigating between elements of the linked data structure in an unlocked state to effect a first set of element state transitions~~dividing the operation into first and second distinct sets of operation tasks~~;

(b) a means for developing a second part of each of the operations operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions~~performing the first set of operation tasks in a first phase~~; and

(c) a means for performing the second part of each of the operations in a second phase distinct from the first phase~~set of operation tasks in a second phase~~.

34. (Cancelled)

35. (Cancelled)

36. (Original) A consistent method of executing simultaneous operations on a linked data structure having at least one element, the method comprising the steps of:

performing any first phase operation task of each of the simultaneous operations in a first phase using parallel processes;

developing a set of serial operations during the first phase; and

performing any second phase operation task of each of the simultaneous operations in a second phase, the second phase operation task including at least one of the set of serial operations.

37. (Original) The method of claim 36 wherein at least one of the simultaneous operations includes an element insertion operation, the first phase operation task of the element insertion operation being performed on an unlocked portion of the linked data structure.

38. (Original) The method of claim 36 wherein at least one of the simultaneous operations includes an element deletion operation, the second phase operation task of the element deletion operation being performed independently of navigation of the linked data structure.

39. (Original) The method of claim 36 wherein the first phase operation tasks are asynchronous and use existing links to navigate the linked data structure.

40. (Original) The method of claim 36 wherein the first phase operation tasks of more than one of the simultaneous operations are completed before the second phase of any of the simultaneous operations is initiated.

41. (New) A method for executing a plurality of operations upon a linked data structure, the method comprising the steps of:

(a) performing a plurality of first phase operation tasks in a first phase, each of the plurality of operations being associated with a different member of the first phase operation tasks;

(b) developing a plurality of second phase operation tasks during the first phase, each of the plurality of operations being associated with a different member of the plurality of second phase operations; and

(c) performing the plurality of second phase operations tasks in a second phase after completion of the first phase.

42. (New) The method of claim 41, wherein the first phase operation tasks are operable to effect a first set of element state transitions and the plurality of second phase operation tasks are

operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions.

43. (New) The method of claim 41, wherein developing the plurality of second phase operation tasks includes developing a series of pointers to elements of the linked data structure and storing the series of pointers external to the linked data structure.

44. (New) The method of claim 41, wherein the plurality of first phase operation tasks are performed in parallel on the linked data structure in an unlocked state and the plurality of second phase operation tasks are performed in series.

45. (New) The method of claim 41, wherein the plurality of second phase operation tasks are performed by navigating to a plurality of elements within the linked data structure using a plurality of pointers external to the linked data structure.

46. (New) A computer readable medium for executing insertion and deletion operations upon a linked data structure having at least one element, the computer readable medium comprising:

(a) a code segment for performing a first set of operation tasks in a first phase for each insertion and deletion operation, the first set of operation tasks operable to effect a first set of element state transitions;

(b) a code segment for developing a second set of operation tasks for each insertion and deletion operation, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and

(c) a code segment for performing the second set of operation tasks in a second phase.

47. (New) A method for executing operations upon a linked data structure having at least one element, the method comprising the steps of:

(a) performing a first part of each of the operations in a first phase by navigating between elements of the linked data structure in an unlocked state, the first part of each of the operations being configured to effect a first set of element state transitions;

(b) developing a second part of each of the operations operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and

(c) performing the second part of each of the operations in a second phase, the second phase being distinct from the first phase.

48. (New) A method of inserting a plurality of elements into a linked data structure, the method comprising:

in a first phase

navigating the linked data structure to a first insertion point for insertion of a first member of the plurality of elements,

associating the first member of the plurality of elements with the linked data structure, the first member of the plurality of elements being in a pending insert state,

storing a first pointer to the first insertion point in a list external to the linked data structure,

navigating the linked data structure to a second insertion point for insertion of a second member of the plurality of elements,

associating the second member of the plurality of elements with the linked data structure, the second member of the plurality of elements being in a pending insert state,

storing a second pointer to the second insertion point in the list external to the linked data structure, storing the first pointer and storing the second pointer; and

in a second phase separate from the first phase

reading the first pointer from the list,

using the read first pointer to navigate to the first insertion point,

completing the insertion of the first member of the plurality of elements in the linked data structure,

reading the second pointer from the lines,

using the read second pointer to navigate to the second insertion point, and

completing the insertion of the second member of the plurality of elements in the linked data structure.